18. (Twice Amended) A method according to claim 9, wherein said authentication details include data identifying the rights of access of individual users to one or more of said resource servers.

REMARKS

Reconsideration and allowance of this application are respectfully requested. Currently, claims 1-19 are pending in this application.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current Amendment. The attached is captioned "**Version With Markings to Show Changes Made**."

Objections to the Disclosure and Claims:

The disclosure was objected to because of the following informalities: "initialisation" should be "initialization" in the disclosure and "authorised" should be "authorized" in claim 1. Applicant has editorially revised the disclosure and claim 1 in light of the Examiner's helpful suggestions.

Rejections Under 35 U.S.C. §102 and §103:

Claims 1, 3, 4, 8-11, 13 and 15-19 were rejected under 35 U.S.C. §102(e) as allegedly being unpatentable over Levergood et al (U.S. '780, hereinafter "Levergood"). Applicant respectfully traverses this rejection.

For a reference to anticipate a claim, each element must be found, either expressly or under principles of inherency, in the reference. Applicant respectfully submits that Levergood fails to disclose (or even suggest) each element of the claimed invention. For example, Levergood fails to disclose

storing status data indicating an identifier to be a validating identifier of a terminal of a currently authenticated user as required by independent claim 1. Moreover, Levergood further fails to disclose enabling a resource server to validate a document request by checking the status data on receipt of the document request as further required by independent claim 1. Similar comments apply to independent claim 9.

An exemplary embodiment of the claimed invention therefore (i) stores status data indicating an identifier issued to an authenticated user's client terminal as being a validating identifier, and (ii) validates document requests by a resource server by checking the status data. The claimed method therefore involves the check of something other than the received identifier for validity. In particular, the associated status data is checked.

Levergood discloses an authentication server 54 which performs the authentication of a client terminal 50 and then issues a command (the re-direct command) to re-direct the client terminal 50 to a desired content server 52. The re-direct command provides a URL which, in addition to the normal URL of the content server 52, also includes what is referred to as a session identification (SID). (See step 9 in Fig. 3 of Levergood.) Content server 52 may validate the URL/SID when it receives a command from the client terminal 50 for documents.

The system described in Levergood does not store status data in an authentication server, the status data indicating that an identifier is a validated identifier (of a terminal of a currently authenticated user). The system

described in Levergood also does not include a resource server being enabled to validate a document request by checking the state of status data on receipt of the document request. In fact, the system described by Levergood works in a very different way since the content server 52 validates a SID entirely independently of the authentication server 54. This is emphasized at col. 7, lines 64-67 which states "If a valid account exists for the user, an SID is issued which authorizes access to the controlled page 'report' and all the other pages within the domain." Accordingly, this passage of Levergood makes it clear that the SID in itself authorizes access to a controlled page. This authorization is contained in the unforgeable signature in combination with the expiration time. (See col. 8, lines 1-5).

Although in Levergood it is implicit that status data indicating whether a terminal or user thereof is authorized to be issued an identifier or SID, there is nothing in Levergood that discloses storing status data which indicates which users are currently authenticated, and which further indicates that this identifier is validated for such an authenticated user. The storage of such information in the present invention provides the benefit of allowing an identifier to be validated without the need for an unforgeable signature or other self validation data. In the passage noted in the Office Action at col. 3, lines 44-47, no mention is made of validating an SID by checking its status at the authentication server. This is because the status data indicating that the SID is valid is contained in the unforgeable signature and the expiry time, both of

which are contained within the SID itself. Accordingly, the claimed invention is not anticipated by Levergood.

Moreover, there would be no reason for one skilled in the art to modify the system of Levergood in order to arrive at the present invention, since in the system of Levergood, a SID is self validating (by virtue of the unforgeable signature and the expiry time), thereby removing the need for the status of a SID to be checked with reference to an authenticating server. Furthermore, because in the present invention the identifier need not be self validating, all that is required of an identifier is information to identify that it is unique in relation to the other identifiers issued to different users of the system. This advantage is not realized in the system of Levergood since a SID must contain the complete set of information required by the content server to determine whether or not a client terminal is authorized to request a file (see col. 8, lines 1-6).

A further advantage of the present invention is that because an identifier need not include the IP address of a client terminal, the client may be of the type that only has access to the resource server via a proxy server. In such situations, the authentication server will only communicate directly with the proxy server, and cannot use a client terminal's IP address to identify the client terminal. The authentication server is thus only aware of the proxy server's network IP address. This advantage is clearly not suggested by Levergood, since each SID must include the client IP address (see col. 3, line 36, col. 6, lines 13 and 63, and col. 8, line 4).

Accordingly, Applicant respectfully submits that claims 1, 3, 4, 8-11, 13 and 15-19 are not anticipated by Levergood and respectfully requests that the rejection of these claims under 35 U.S.C. §102 be withdrawn.

Claim 2 was rejected under 35 U.S.C. §103 as allegedly being unpatentable over Levergood in view of Kirsch (U.S. '915). Claims 5-7, 12 and 14 were rejected under 35 U.S.C. §103 as allegedly being unpatentable over Levergood in view of See et al (U.S. '243, hereinafter "See"). Applicant respectfully traverses these rejections.

Since claims 2, 5-7, 12 and 14 depend from one of independent claims 1 and 9, all of the comments made above with respect to Levergood apply equally to these claims. Neither Kirsch nor See remedies the above deficiencies of Levergood discussed above with respect to the claimed invention. Accordingly, even if Kirsch or See were combined with Levergood, the resulting combinations would not have taught or suggested all of the claimed limitations. Accordingly, Applicant respectfully requests that the rejections of claims 2, 5-7, 12 and 14 under 35 U.S.C. §103 be withdrawn.

Conclusion:

Applicant believes that this entire application is in condition for allowance and respectfully requests a notice to this effect. If the Examiner has any questions or believes that an interview would further prosecution of this application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____
Raymond Y. Mah
Reg. No. 41,426

RYM/sl
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703)816-4044
Facsimile: (703)816-4100

IN THE SPECIFICATION:

Paragraph beginning at page 2, line 5 has been amended as follows:

In accordance with one aspect of the invention there is provided a method of operating an authenticating server system for authenticating users at client terminals connected via a data communications network, to control access to a document stored on a resource server, said method comprising performing the following steps in said server system:

storing authentication details of [authorised] authorized users:

Paragraph beginning at page 4, line 28 has been amended as follows:

The SPS has an associated data store 8 which holds authentication details for each of the users [authorised] authorized to have access to the application servers APS and a token identifying the access rights of each user. The CMS has an associated data store 10, which holds details of users currently logged on for access to the application server APS, and which maintains logging on histories for users once they are logged off.

Paragraph beginning at page 8, line 3 has been amended as follows:

At this point, the user is fully logged-on in a CMS, with the log on notification details stored in the CMS store 10. The user may then, via an appropriate application client APC access one [ore] or more of the application servers APS which the user is [authorised] authorized to access, as specified by the access right token.

Paragraph beginning at page 10, line 25 has been amended as follows:

The authentication scheme described in relation to users at terminals T1 or T2 described above involves identification of a user, after initial authentication, by the IP address of the terminal at which the user is logged on. Because the CMS performs periodic re-authentication of the user, it is difficult for a third party to impersonate the user by IP address spoofing. Namely, even if a third party were to spoof the IP address of the user, the third party would only have access to the real user's resources for the time provided by the user's timer in the CMS. Once the timer has expired, the third party forming the IP spoofing would not be able to re-authenticate, without access to the user's password. Since the user's password is only ever sent across the Internet when a password change occurs, and even then in encrypted form, a third party has no means of finding out the password of an [authorised] authorized user.

Paragraph beginning at page 12, line 24 has been amended as follows:

In return, the APS receives the authentication details from the APC, step 84, including the same address token, whereby the user is re-identified, and the username and password, on which the SPC performs the first hashing function H0 illustrated in Figure 4. This information is passed on to the CMS, which polls the SPS to check whether the username and password hash matches one stored in the SPS store 8 as that of an [authorised] authorized user.

Paragraph beginning at page 17, line 31 has been amended as follows:

Each data block includes five parts, including an [initialisation] initialization vector 150 for the decryption process, added during encryption and prior to transmission of the block. The block also includes a block number, 152, which increments with each block of data sent, and a data count 154, which is a count of the number of data bytes included in the data block, excluding the [initialisation] initialization vector 150, block number, data count, checksum and any padding added during the encryption process. The next part of the data block is the part holding the encrypted data 156, which is padded to a multiple of 8 bytes by the encryption function if the data block is not otherwise a multiple of 8 bytes. The final part of the data block is an encryption checksum 158, which is added by the encryption function and checked and removed by the decryption function to ensure that the data block has been received correctly after transmission.

IN THE CLAIMS:

1. (Amended) A method of operating an authenticating server system for authenticating users at client terminals connected via a data communications network, to control access to a document stored on a resource server, said method comprising performing the following [steps] in said server system:

storing authentication details of [authorised] authorized users;

receiving authentication data for a user from a client terminal of the user and validating said authentication data by reference to said stored authentication details;

issuing an identifier for the user's client terminal to said terminal for storage thereon, the identifier being transmitted in such a manner that the identifier is retransmitted by said [user] user's client terminal with document requests directed at said resource server;

storing status data indicating said identifier to be a validated identifier of a terminal of a currently authenticated user, in response to the receipt and validation of the authentication data [said authentication step]; and

enabling said resource server to validate a request for said document from the user's client terminal, which request includes said identifier, by checking said status data on receipt of said document request.

2. (Amended) A method according to claim 1, wherein said identifier is transmitted in a cookie to said [user] user's client terminal.

3. (Twice Amended) A method according to claim 1, wherein [said authentication step comprises receiving] said identifier is received from said [user] user's client terminal with said authentication data.

4. (Amended) A method according to claim 3, wherein [said authentication step comprises issuing] a new identifier is issued to said [user] user's client terminal if said authentication data is invalid.

5. (Amended) A method according to claim 4, wherein said identifier comprises data indicating the number of times an invalid authenticator has been received from said [user] user's client terminal.

6. (Amended) A method according to claim 5, wherein said method comprises issuing no further identifier to said [user] user's client terminal if an identifier received from said [user] user's client terminal indicates that a predetermined number of invalid authenticators have been received from said [user] user's client terminal.

7. (Twice Amended) A method according to claim 1, comprising timing out said identifier as an identifier of a terminal of a currently authenticated user if no document request is received from said [user] user's client terminal for a predetermined period.

8. (Twice Amended) A method according to claim 1, comprising authenticating said user for access to a plurality of Web servers located in the same Internet domain; and

enabling each of said Web servers to validate document requests from the user's client terminal, which requests include said identifier, by checking said status data on receipt of a document request.

9. (Amended) A method of operating an authenticating server system for authenticating users at client terminals remotely connected via a data communications network, to control access to a plurality of resource servers, said method comprising performing the following steps in said server system:

storing authentication details of [authorised] authorized users;

performing remote authentication of a user by reference to said stored authentication details and during said remote authentication step generating status data, distinguishing said user from other users which are not currently authenticated, and a secret encryption key shared with said user;

storing said status data in storage means accessible to said plurality of resource servers to check an authentication status of said user by using an identifier for the user's client terminal received in a service request; and

storing said shared secret key in a data store accessible by at least one of said resource servers for use during communications with said user.

10. (Amended) A method according to claim 9, wherein said remote authenticating step comprises issuing a challenge to the user's client terminal, receiving a response to said challenge, and verifying said response.

14. (Twice Amended) A method according to claim 12, wherein said updating step is performed in response to a request by the user's client terminal.

15. (Twice Amended) A method according to claim 9, wherein said identifier is an IP address of the user's client terminal.

16. (Twice Amended) A method according to claim 9, wherein said authentication step comprises issuing said identifier to the user's client terminal.

18. (Twice Amended) A method according to claim 9, wherein said authentication details include data identifying the rights of access of individual users to one or more of said [application] resource servers.